ONLINE HANDWRITING GESTURE RECOGNITION USING TRANSFORMER AND NATURAL LANGUAGE PROCESSING

Guénolé C.M. Silvestre[†], *Félix Balado*[†], *Olumide Akinremi*[†] *and Mirco Ramo*[†,‡]

[†] School of Computer Science, University College Dublin, Ireland
[‡] Dip. Ingegneria dell'Informazione, University of Pisa, Italy

ABSTRACT

The Transformer architecture is shown to provide a powerful machine transduction framework for online handwritten gestures corresponding to glyph strokes of natural language sentences. The attention mechanism is successfully used to create latent representations of an end-to-end encoder-decoder model, solving multi-level segmentation while also learning some language features and syntax rules. The additional use of a large decoding space with some learned Byte-Pair-Encoding (BPE) is shown to provide robustness to ablated inputs and syntax rules. The encoder stack was directly fed with spatio-temporal data tokens potentially forming an infinitely large input vocabulary, an approach that finds applications beyond that of this work. Encoder transfer learning capabilities is also demonstrated on several languages resulting in faster optimisation and shared parameters. A new supervised dataset of online handwriting gestures suitable for generic handwriting recognition tasks was used to successfully train a small transformer model to an average normalised Levenshtein accuracy of 96% on English or German sentences and 94% in French.

Index Terms— Online Gesture Recognition, Transformer, Multilevel Segmentation, Language Models, Transfer Learning, Multi-head Attention.

1. INTRODUCTION

Handwriting Character Recognition (HCR) when associated with touch-sensitive display panels provides an intuitive and seamless input mechanism eschewing the need for structured UIs such as virtual keyboards, often slow and error-prone while also distant to the natural handwriting experience.

In this context, online gesture recognition of glyphs refers to the problem of mapping a set of user gestures corresponding to sequences of spatio-temporal samples into their corresponding symbolic representation. Each n-dimensional sample individuates a touch. A coherent and consecutive sequence of touches defines a stroke that can be combined to form glyphs. Glyphs correspond to characters or symbols encoded in a language vocabulary. It also extends to a wider context of symbols drawn from large alphabets composed of glyphs with many strokes such as logographic systems or mathematical expressions. Table 1 formalises the terminology adopted in this work.

These applications must jointly solve a number of tasks, namely i) accurate feature extraction in a multi-dimensional spatio-temporal space, ii) segmentation of stroke sequences to identify glyphs, iii) glyph segmentation for the purpose of word/sentence recognition, and iv) the encoding of syntax rules and language patterns to form a correct symbolic output. An example of an online gesture sequence is depicted in Fig. 1.

Table 1. Terminology

Term	**Definition**
touch/ point	$(x, y, [t, p])$ location on touch panel sampled at t with finger pressure p
stroke	Sequence of points where finger consecutively touches the panel
glyph	List of one or more strokes individuating an element in the vocabulary
symbol	Item in a vocabulary of 57 symbols $\{\mathtt{a}-\mathtt{z}, \mathtt{A}-\mathtt{Z}, \text{␣}, \langle\mathtt{unk}\rangle, \langle\mathtt{bos}\rangle, \langle\mathtt{eos}\rangle, \langle\mathtt{pad}\rangle\}$ or 2 000 BPE tokens trained from $\{\mathtt{en}, \mathtt{fr}, \mathtt{de}\}$ datasets

2. RELATED WORK & CONTRIBUTIONS

The field of HCR consists of techniques aiming at generating text directly from handwritten inputs. Most solutions rely on offline data due to dataset availability [1,2]. However, the temporal dimension provides some valuable information that may simplify stroke segmentation, avoiding recourse to convoluted regression strategies such as text-line segmentation [3]. As a result, online methods generally exhibit superior performance over offline counterparts as reported in [4–6]. With the growing popularity of the attention mechanism [7,8], the field remains in constant development with much effort and resources devoted to improving existing techniques [9,10]. In a related sub-problem, Handwritten Mathematical Expression Recognition (HMER) consists in the generation of mathematical expressions using formal syntaxes, with state-of-the-art HMER models reaching impressive levels of accuracy, particularly when exploiting attention [11] and combining modalities from online and offline data [12]. However, these models fail to learn the intrinsic structure of expressions. This is somewhat addressed in [13] using an RNN encoder along with a syntactic tree decoder.

In the context of sequence transduction tasks [14], the Transformer [15] framework stands as the state-of-the-art on almost all NLP tasks [16–19], eschewing recourse to recurrent or convolutional units. This powerful sequence mapping architecture relies entirely on the attention mechanism [7], allowing for significant parallelisation and unattenuated gradient flow. It has also been successfully applied to a wider and more generic group of sequence transduction problems [20–24]. The Transformer popularity saw many proposals to revisit and optimise its design [25–28] but very few are capable of clearly outperforming the original topology.

This work follows the seminal work by [15] and proposes to reformulate the online gesture recognition problem as a neural transduction task, leveraging the power of attention in the context of natural languages.

Main Contributions

(i) New online datasets are proposed for handwritten text in natural languages (cf. Section 3) and suitable for a wide range of supervised and unsupervised machine learning applications.

(ii) The attention mechanism is shown to successfully learn and represent implicit structures of spatio-temporal gesture data.

(iii) The power of transformers is demonstrated not only as language models but also as a solution to several sequence mapping tasks, with transfer learning behaviours observed for the encoder[1].

(iv) A small footprint[2] topology is proposed as an end-to-end model, with fast optimisation, high accuracy and suitable for edge inference.

(v) Model robustness is demonstrated on ablated inputs with the ability to generate grammatically compliant expressions in case of missing strokes.

(vi) Multi-level segmentation capability is highlighted in the correlation between syntactically correct predictions and their explainability in cross-attention visualisation.

3. DATASETS

An important contribution of this work is that of online gesture datasets for text in English (en), French (fr) or German (de), suitable for investigating the task of HCR but also segmentation of touch, stroke or glyph, and eventually grammatical/syntactical compliance of expressions in natural languages. Our handwritten database is presented as a coherent collection of tables composing a relational schema with spatio-temporal data for Roman alphabets, Arabic numerals [6], mathematical and punctuation symbols, collected from volunteers writing on touch panels. This stage saw the contribution of over 600 subjects for a total of 69 278 labelled glyphs composed by 93 330 strokes, with over 2 million touches. The dataset can be used at different levels of granularity, namely *touch*, *stroke* and *glyph*. In this work, we report results at the stroke level, leaving the burden of glyph segmentation to the model.

Subjects have been split into training, validation and test sets (60/20/20 proportions) such that models were tested on unseen handwriting styles to ensure accurate estimation of the generalisation power. When online HCR was performed in the context of natural languages, the WMT19 news dataset [29] was used to generate sequences of spatio-temporal gestures from our relational schema for all available sentences ($\simeq$ 100 000 sentences chosen to be no longer than 200 input strokes). Pre-processing was applied to substitute or omit characters where gesture data was unavailable e.g. {é, è, ê} → e in fr or ß → ss in de. While these datasets are much smaller than those used to train large NLP models [30], it was found to be sufficient to demonstrate the model's ability to successful learn some language features at the sentence level.

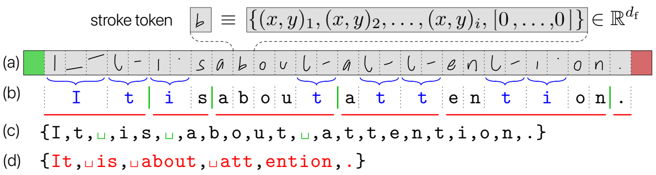

stroke token
$\equiv \{(x,y)_1, (x,y)_2, \ldots, (x,y)_i, [0,\ldots,0]\} \in \mathbb{R}^{d_\mathrm{f}}$
(a)
(b)
I t i s a b o u t a t t e n t i o n .
(c) {I,t,␣,i,s,␣,a,b,o,u,t,␣,a,t,t,e,n,t,i,o,n,.}
(d) {It,␣is,␣about,␣att,ention,.}

Fig. 1. Online gesture example of an input stroke sequence (a) for the sentence (b) and its corresponding output list – symbols in (c) and BPE tokens in (d). Cells in (a) depict the linear interpolation of spatio-temporal points that forms input tokens. Green and red cells denote the ⟨bos⟩ and ⟨eos⟩ input token respectively. Stroke sequence segmentation, glyph segmentation and parsing of language tokens are colour coded with with blue, green and red, respectively.

4. TRANSFORMER ARCHITECTURE & EXPERIMENTAL DETAILS

Our model [31] leverages the original Transformer [15] architecture. However crucial modifications are introduced to work with spatio-temporal data. Given some input sequence, $X \in \mathbb{R}^{d_\mathrm{f} \times n}$, of n stroke tokens defined as interleaved spatio-temporal data with zero-padding of fixed-length d_f —appropriately prefixed, suffixed and padded at the end with ⟨eos⟩, ⟨bos⟩ and ⟨pad⟩ tokens respectively —, a mask M_x is computed to ensure encoder's attention is only paid to valid online data tokens. Here, $d_\mathrm{f} = 64 \times 2$ allowing for a maximum of 64 (x, y) touch samples per stroke.

As the input is composed of spatio-temporal information corresponding to touches[3], each encoder token embeds a stroke as d_f scalars (cf. Fig. 1) resulting in the identification within a potentially unbounded input vocabulary and therefore eschewing any form of embedding.

Positional encoding provides a strategy to embed the positional information of input tokens in the encoder, a necessary operation since the attention mechanism has no built-in concept of sequentiality. Frequency modulation is proposed in [15]. However, since we observed no performance gain with such a strategy, we use a learnable 1D embedding based on the token index. Stroke positions are encoded in $P_x \in \mathbb{R}^{d_\mathrm{f} \times n}$.

The encoder is trained to learn some latent sequence representation $Z = \mathrm{Enc}(X + \alpha P_x, M_x) \in \mathbb{R}^{d_\mathrm{a} \times d_\mathrm{h} \times n}$ where α is a scaling factor blending input data and positional information, d_a the number of attention heads (2 to 4) and d_h the hidden state dimension of the attention heads. The encoder con-

[1] Encoder with frozen parameters pre-trained on English word dataset can be used on other languages with alphabets using some potentially unknown glyphs.

[2] Despite its small size, model can perform the tasks of glyph segmentation, character recognition, word segmentation and sentence construction at remarkable performance levels, learning efficiently the input/output mapping.

[3] Explicit sampling time and pressure (t, p) features are observed to have no impact on performance suggesting some level of information redundancy.

sists of a stack of l_e identical multi-head vanilla self-attention layers and a positional feed-forward network of hidden dimension d_p. Each layer is followed by a residual connection before layer-normalisation.

When exploring transfer learning capabilities, an encoder with $\Theta_\mathrm{e} = 523\,520$ parameters is used as a feature extractor resulting in a considerable speed-up during training and model optimisation.

The decoder generates a causal sequence of tokens in an auto-regressive manner given some vocabulary and relative token encoding. It is initialised with the $\langle$`bos`$\rangle$ token and iteratively outputs a new token using greedy sampling of the decoder's softmax output until the $\langle$`eos`$\rangle$ token is predicted or the maximum sequence length, m, is reached. The decoder also consists of l_d identical layers each composed by: i) a masked self-attention layer (with d_a attention heads of hidden state dimension d_h) that prevents the decoder from peeking at the subsequent tokens, ii) a cross-attention layer that attends over the encoder output Z to generate predictions, and iii) a feed-forward layer just as in the encoder but of dimension $k\, d_\mathrm{p}$ where $k=1$ to 3.

At each step, the decoder's input, $Y_{<t}$, is an auto-regressive sequence of tokens adequately masked with M_y and used to predict the next token of the output sequence: $Y_t = \mathrm{Dec}(Y_{<t}, M_y, Z)$. All Θ_d parameters of the decoder were trained from some randomly initialised state.

Experimental details: All models were configured with $d_\mathrm{f} = d_\mathrm{a} \times d_\mathrm{h} = d_\mathrm{p} = 128$. For $\mathrm{v}_{61}{}^{\texttt{rnd}}$–$\mathrm{v}_{68}{}^{\texttt{gow}}$, $n = 2m = 48$ and $k=1$. For $\mathrm{v}_{74}{}^{\texttt{en}}$–$\mathrm{v}_{93}{}^{\texttt{de}}$, $n=2\,m=200$ and $k=3$. Parameters of the attention layer stack are detailed Tables 2 and 3. Models were trained on Nvidia TitanX GPUs[4], for a maximum of 5000 epochs, a 256 batch size, using cross-entropy loss and Adam optimiser with a decay schedule (initial learning rate of 8×10^{-4} and halving every 30 epochs).

5. EXPERIMENTAL RESULTS

A series of experiments were carried out to determine suitable model hyper-parameters and investigate the benefits of increasing the decoder's output dimension using vocabularies (vocabs) of various sizes: i) a small vocab of 57 alphabetic symbols, and ii) larger vocabs with 2 000 Byte-Pair-Encoding (BPE) symbols trained on three natural languages (`en`, `fr`, `de`). Models were evaluated using a number of performance metrics on the test sets and results are reported in terms of Cross-Entropy Loss (XEL) and two edit distance metrics, namely: normalised Levenshtein distance [32] Accuracy (LA) and Character Error Rate (CER).

Hyper-parameters were determined from XEL results obtained on models trained with pseudorandom letter words as reported in Table 2. Since this only required to solve the tasks of glyph segmentation and classification, splitting hidden states in two multi-attention heads was sufficient. A five-layer encoder abstraction performed best, a result consistent with previous observations in an online classification context using convolutional topologies [6]. Note that there are no benefits in increasing the decoder's layer abstraction since output tokens are i.i.d. in a small 57 symbol vocab, with no symbol patterns to be exploited.

Table 2. Hyper-parameter search for models trained on stroke sequence corresponding to pseudorandom letter words. Performance is reported in terms of Cross-Entropy Loss (XEL).

Name	Enc $(l_\mathrm{e}, d_\mathrm{a})$	Dec $(l_\mathrm{d}, d_\mathrm{a})$	$\Theta_\mathrm{e} + \Theta_\mathrm{d}$	XEL
$\mathrm{v}_{61}{}^{\texttt{rnd}}$	2L, 4H	2L, 4H	554 552	0.761
$\mathrm{v}_{62}{}^{\texttt{rnd}}$	3L, 2H	2L, 2H	654 136	0.881
$\mathrm{v}_{63}{}^{\texttt{rnd}}$	4L, 4H	4L, 4H	1 085 496	0.914
$\mathrm{v}_{64}{}^{\texttt{rnd}}$	**5L, 2H**	**2L, 2H**	850 232	**0.690**
$\mathrm{v}_{65}{}^{\texttt{rnd}}$	5L, 2H	4L, 2H	1 185 080	1.556
$\mathrm{v}_{66}{}^{\texttt{rnd}}$	7L, 2H	2L, 2H	1 052 472	0.895
$\mathrm{v}_{67}{}^{\texttt{rnd}}$	7L, 2H	4L, 2H	1 384 248	1.332

Table 3. Model performance reported in term of Cross-Entropy Loss (XEL), normalised Levenshtein distance Accuracy (LA) and Character Error Rate (CER). $\mathrm{v}_{68}{}^{\texttt{gow}}$ trained on a group of `en` words (57-symbol vocab), $\mathrm{v}_{80}{}^{\texttt{en}}$–$\mathrm{v}_{93}{}^{\texttt{de}}$ trained on sentences in three languages (2 000 BPE vocab). $\Theta_\mathrm{e} = 523\,520$ (all), $\Theta_\mathrm{d} = 330\,041$ ($\mathrm{v}_{68}{}^{\texttt{gow}}$) and 1 453 520 (others). Fine-tuned models provide best performance.

Name	Enc $(l_\mathrm{e}, d_\mathrm{a})$	Dec $(l_\mathrm{d}, d_\mathrm{a})$	XEL	LA(%)	CER
$\mathrm{v}_{68}{}^{\texttt{gow}}$	5L, 4H‡	2L, 4H‡	0.282	91.07	0.089
$\mathrm{v}_{74}{}^{\texttt{en}}$	5L, 4H‡	4L, 4H‡	0.260	94.04	0.066
$\mathrm{v}_{80}{}^{\texttt{en}}$	5L, 4H*	4L, 4H‡	0.183	**95.90**	0.045
$\mathrm{v}_{82}{}^{\texttt{fr}}$	5L, 4H*	4L, 4H‡	0.313	93.21	0.074
$\mathrm{v}_{83}{}^{\texttt{de}}$	5L, 4H*	4L, 4H‡	0.285	**94.79**	0.057
$\mathrm{v}_{92}{}^{\texttt{fr}}$	5L, 4H†	4L, 4H†	0.293	**93.51**	0.071
$\mathrm{v}_{93}{}^{\texttt{de}}$	5L, 4H†	4L, 4H†	0.298	94.52	0.060

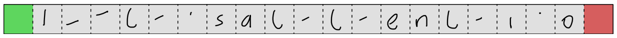
‡Trained from scratch from some randomly initialised state
†Fined-tuned parameters using transfer learning of $\mathrm{v}_{74}{}^{\texttt{en}}$ encoder
*Frozen parameters using transfer learning of $\mathrm{v}_{74}{}^{\texttt{en}}$ encoder

Table 3 summarises the main results with hyper-parameter configuration and performance evaluation for models trained on natural languages. Since this requires solving multilevel segmentation tasks, the hidden states of a 4-layer decoder abstraction were split into 4 attention heads. Model $\mathrm{v}_{68}{}^{\texttt{en}}$ was trained on short `en` sentence chunks to output symbols using the 57 symbol vocab, with some modest data augmentation strategies using affine transformations. Despite its smaller size, model demonstrates strong end-to-end recognition capabilities, learning some syntax rules. Larger models ($\Theta_\mathrm{e}+\Theta_\mathrm{d}=1.9\,\mathrm{M}$) were subsequently trained on full sentences in different languages. $\mathrm{v}_{74}{}^{\texttt{en}}$ was optimised from a random state with the same output configuration, exhibiting some improved performance over the smaller $\mathrm{v}_{68}{}^{\texttt{gow}}$ model. Its encoder parameters were then used as a feature extractor when training larger models outputting BPE tokens (`en`, `fr`, `de`) with a significant optimisation speed-up. With a performance reaching a LA of 96% (4.5% CER, 14.7% WER), the `en` model's improvement is attributed to the larger decoder output dimension that provides some error correction mechanisms as confirmed in the robustness analysis of Table 4, eliminating some spelling errors. Although the encoder was optimised on inputs associated with `en` words, the `de` model $\mathrm{v}_{83}{}^{\texttt{de}}$ fell just 0.4% short of the `en` model accuracy, with strong performance also observed with the `fr` model $\mathrm{v}_{82}{}^{\texttt{fr}}$ tailing by 1.5%. It is worth noting that fine-tuning of the `fr` model over a couple of epochs resulted in modest performance gains.

[4] Nvidia is acknowledged for the donation of GPUs

Table 4. Robustness to elided input, spelling errors, and missing punctuation. BPE models can infer correct expressions.

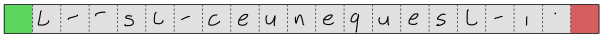
Input (X) for glyph symbols (S) and inference output ($\hat{Y}$)

$X=$ 

$S=\{\texttt{I,t,',s,a,t,t,e,n,t,i,o}\}$

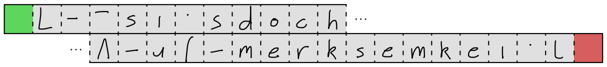
$\mathrm{v}_{68}{}^{\mathrm{en}} \rightarrow \hat{Y}=\{\texttt{I,t,␣,i,s,␣,a,t,t,e,n,t,i,o,n}\}$

$\mathrm{v}_{80}{}^{\mathrm{en}} \rightarrow \hat{Y}=\{\texttt{It,␣is,␣att,ention,.}\}$ ①

$X=$ 

$S=\{\texttt{E,s,t,c,e,u,n,e,q,u,e,s,t,i}\}$

$\mathrm{v}_{82}{}^{\mathrm{fr}} \rightarrow \hat{Y}=\{\texttt{Est,-,ce,␣une,␣question,␣?}\}$ ②

$X=$ 

$S=\{\texttt{E,s,i,s,d,o,c,h,A,u,f,m,e,r,k,s,e,m,k,e,i,t}\}$

$\mathrm{v}_{83}{}^{\mathrm{de}} \rightarrow \hat{Y}=\{\texttt{Es,␣ist,␣doch,␣Auf,mer,k,sam,keit,.}\}$ ③

Robustness and Visualisation

Model robustness is investigated with input stroke ablation and deliberate erroneous input while observing the model's ability to enforce syntax rules and correct spelling.

As all dataset expressions end with some punctuation mark, the learning of this rule is observed in all models as shown in Table 4. In addition, interrogative forms are detected from the subject/verb inversion, with correct insertion of a question mark as shown in the `fr` example ②. One can also note the correct hyphenation token inferred between verb and subject pronoun as expected in `fr` syntax, along with spacing before the question mark. The large BPE vocab also provides some robustness to spelling errors as shown in the `de` example ③. This was not observed to the same extent on models with smaller output vocab such as $\mathrm{v}_{68}{}^{\mathrm{en}}$. Finally, contraction of `en` verbs is correctly detected and expanded as seen in the `en` example ①. These observations demonstrate that the models are capable of learning non-trivial valuable syntax and grammar rules along with some language features despite the small dataset and model size.

Visualisation of the attention mechanisms provides some interesting insights in the learning process. Fig. 2 depicts weights of the decoder's cross-attention over the encoder's output. It shows that head 3 of layer 4 is key to token segmentation with strong attention paid to the first token stroke solving the word segmentation task. The large BPE space is also leveraged for the auto-completion of the final text token, while insertion of the punctuation mark is carried out by tracking the final stroke. The ⟨`eos`⟩ token output is seen focusing on both final stroke and ⟨`eos`⟩ stroke input.

6. CONCLUSION

This work posed the problem of online HCR as a transduction task, using a Transformer framework to learn the complex mapping of online input gesture data corresponding to handwritten strokes of natural language sentences. The encoder's input was modified to receive spatio-temporal data as real-valued tokens, operating at stroke level without the need for mapping on a fixed input vocabulary. Models were shown to predict text with very high accuracy by handling internally the multi-level segmentation of inputs (at glyph and word levels), and also understanding and learning how to represent and enforce syntactic and semantic rules of data. Index positional encoding was shown to be as effective as cosine modulation yet standing as a simpler and more natural encoding for the position information. Although the Transformer's ability to generate complex representations and learn non-trivial input/output mapping between sequences is well established [16, 20], the challenge was further pushed in this work in the absence of ad hoc syntax, semantic rules, or engineered loss computation and architecture.

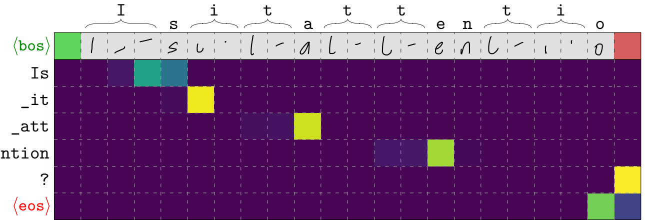


Fig. 2. Cross-attention plot of head 3–layer 4 for model $\mathrm{v}_{80}{}^{\mathrm{en}}$ showing output tokens tracking its first stroke. Question mark token is focusing on input 🟥 token.

In addition, an encoder trained on a specific language was successfully used as a frozen feature extractor in the optimisation of decoders in several other language domains. Such transfer learning capabilities suggest that pre-trained encoders can create general latent representations suitable for problems of different nature, resulting in model size reduction, training acceleration with no need for fine-tuning or explicit domain adaptation. This will also benefit applications where computational power/time and dataset size are limited.

The objective of this work was not so much to push out some state-of-the-art model but rather to state some important considerations that may be the starting points for future works on sequentiality of other signal types. Neural transduction may be extended in this way to online data at different granularity levels, with no need for separate input segmentation or complex positional embeddings. With larger language datasets and computational resources, this approach may reveal deeper language modelling capabilities to similar levels observed in BERT or GPT [16,30], and this straight from digital signals. The end-to-end encoder-decoder models in this work achieved a normalised Levenshtein accuracy of 94% to 96% at sentence levels on the three languages considered and directly from online data.

7. REFERENCES

[1] R. Plamondon and S.N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[2] Deepak Sinwar, Vijaypal Singh Dhaka, Pradhan, et al., "Offline script recognition from handwritten and printed multilingual documents: a survey," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 24, no. 1, pp. 97–121, 2021.

[3] Berat Barakat, Ahmad Droby, Majeed Kassis, and Jihad El-Sana, “Text line segmentation for challenging handwritten document images using fully convolutional network,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 374–379.

[4] JinHyung Kim and Bong-Kee Sin, *Handbook of Document Image Processing and Recognition*, chapter Online Handwriting Recognition, pp. 887–915, Springer London, 2014.

[5] Anchit Shrivastava, Isha Jaggi, Gupta, et al., “Handwritten digit recognition using machine learning: A review,” in *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, 2019, pp. 322–326.

[6] Philip J Corr, Guenole C Silvestre, and Chris J Bleakley, “Open source dataset and deep learning models for online digit gesture recognition on touchscreens,” in *2017 Irish Machine Vision and Image Processing Conference (IMVIP)*, 2017.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” in *International Conference on Learning Representations (ICLR)*, Jul 2015.

[8] Jason Poulos and Rafael Valle, “Character-based handwritten text transcription with attention networks,” *Neural Computing and Applications*, vol. 33, no. 16, pp. 10563–10573, 2021.

[9] Daniel Keysers, Thomas Deselaers, Rowley, et al., “Multi-language online handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180–1194, 2017.

[10] Alex Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint*, `https://doi.org/10.48550/arXiv.1308.0850`, 2013.

[11] Zhe Li, Lianwen Jin, Lai, et al., “Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention,” in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 175–180.

[12] Jiaming Wang, Jun Du, Zhang, et al., “Multi-modal attention network for handwritten mathematical expression recognition,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1181–1186.

[13] Jianshu Zhang, Jun Du, Yang, et al., “SRD: A tree structure based decoder for online handwritten mathematical expression recognition,” *IEEE Transactions on Multimedia*, vol. 23, pp. 2471–2480, 2021.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, “Sequence to sequence learning with neural networks,” *arXiv preprint*, `https://doi.org/10.48550/arXiv.1409.3215`, 2014.

[15] Ashish Vaswani, Noam Shazeer, Parmar, et al., “Attention is all you need,” in *Advances in neural information processing systems*, 2017, vol. 30, pp. 6000–6010.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Jun 2019, pp. 4171–4186, Association for Computational Linguistics.

[17] Tom Brown, Benjamin Mann, Ryder, et al., “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[18] Thomas Wolf, Lysandre Debut, Sanh, et al., “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[19] Ankush Garg and Mayank Agarwal, “Machine translation: a literature review,” *arXiv preprint*, `https://doi.org/10.48550/arXiv.1901.01122`, 2018.

[20] Niki Parmar, Vaswani, et al., “Image transformer,” in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[21] Cheng-Zhi Anna Huang, Ashish Vaswani, et al., “Music transformer: Generating music with long-term structure,” in *International Conference on Learning Representations (ICLR)*, 2019.

[22] Hengshuang Zhao, Li Jiang, Jia, et al., “Point transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16259–16268.

[23] Alexander Kozlov, Vadim Andronov, and Yana Gritsenko, “Lightweight network architecture for real-time action recognition,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 2074–2080.

[24] Andrea D’Eusanio, Alessandro Simoni, Pini, et al., “A transformer-based network for dynamic hand gesture recognition,” in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 623–632.

[25] Sinong Wang, Belinda Z Li, Khabsa, et al., “Linformer: Self-attention with linear complexity,” *arXiv preprint*, `https://doi.org/10.48550/arXiv.2006.04768`, 2020.

[26] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya, “Reformer: The efficient transformer,” in *International Conference on Learning Representations*, 2020.

[27] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, et al., “Rethinking attention with performers,” in *International Conference on Learning Representations*, 2021.

[28] Roshan M Rao, Jason Liu, Verkuil, et al., “MSA transformer,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8844–8856.

[29] Wikimedia Foundation, “Shared task (WMT19): Machine translation of news,” ACL 2019, `http://www.statmt.org/wmt19/translation-task.html`, 2019.

[30] Alec Radford, Jeffrey Wu, Rewon Child, et al., “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, pp. 9–20, 2019.

[31] Olumide Akinremi, “Is this all about attention?,” M.S. thesis, University College Dublin, May 2021.

[32] Li Yujian and Liu Bo, “A normalized Levenshtein distance metric,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.